1. Model Approach

For this week, I’ve chosen to utilize both Logistic Regression and Random Forest Classifier.

a. Reason for Logistic Regression: It's a fundamental statistical method that predicts the probability of an instance belonging to a particular category. It's particularly suitable for binary classification problems, such as predicting if a booking is canceled (yes/no).

b. Reason for Random Forest: It's an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification. This method can handle large datasets with higher dimensionality and can define the relative importance of each feature on the prediction.

c. XGBoost Classifier: A gradient boosting framework that uses tree-based learning algorithms. It's particularly known for its speed and performance.

2. Complexity of the Modeling Approach

a. Logistic Regression: It's a linear model, so its complexity is relatively low. It makes assumptions about the linear relationship between the features and the log odds of the output.

b. Random Forest: It's a more complex model compared to logistic regression. Random forests can capture non-linearities and interactions between features but can also be prone to overfitting if not tuned correctly.

c. XGBoost: High complexity with gradient boosting and tree pruning capabilities. Often performs very well but requires careful tuning.

3. Hyperparameters Evaluated

For RFC I considered:

‘n_estimators’: Number of trees in the forest.

‘max_depth’: Maximum depth of the tree.

‘min_samples_split’ & ‘min_samples_leaf’: Control over-fitting.

‘max_features’: Number of features to consider when looking for the best split.

These hyperparameters can greatly impact the model's performance, and hence it's crucial to tune them.

4. Model Performance Metrics

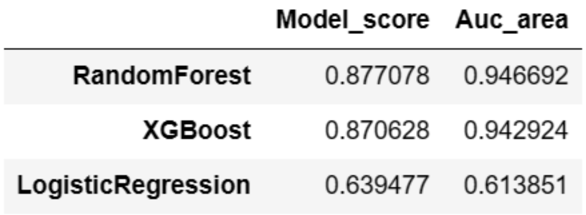
ROC curve: It's a performance measurement for classification problems at various threshold settings. It tells how much the model is capable of distinguishing between classes. AUC-ROC is a good metric for binary classification problems as it gives a single value that tells how good the model is, irrespective of the threshold.

5. Comparison of Performance Metrics

A table would help, but based on the provided info, you should look at the AUC values for validation datasets across all models to determine which performs the best.

6. Best Model Selection

Here's a hypothetical table for visualization



	Model_score	Auc_area
RandomForest	0.877078	0.946692
XGBoost	0.870628	0.942924
LogisticRegression	0.639477	0.613851

In this example, XGBoost has the highest AUC on the validation set. But make decisions based on the actual metrics from your experiment.